

## Opis

Danych jest  $n$  odcinków, których długości są liczbami całkowitymi dodatnimi, umieszczonymi w tablicy  $T[]$ . Napisz w Javie efektywny program, zawierający metodę `SearchBinFirst(x)`, działającą z pesymistyczną złożonością czasową  $O(\log_2 n)$  i pamięciową  $O(1)$ , zwracającą pierwszą pozycję elementu  $x$  w uporządkowanej tablicy, który wyznaczy liczbę możliwych trójkątów indeksów w tablicy zawierających odcinki, z których można zbudować trójkąt.

Każdy odcinek może występować tylko raz w budowanym trójkącie, choć może być wiele odcinków o tej samej długości. Może się też zdarzyć, że otrzymamy kilka trójkątów o takich samych długościach boków, w takim przypadku liczymy je wszystkie.

Przykładowo dla tablicy  $T[] = [2, 2, 3, 3]$  można zbudować 4 trójkąty, których długości boków występują w tablicy pod indeksami:  $(0, 1, 2)$ ,  $(0, 1, 3)$ ,  $(0, 2, 3)$  i  $(1, 2, 3)$ .

W komentarzu, w opisie rozwiązania uzasadnij złożoność obliczeniową rozwiązania.

## Wejście

Dane do programu wczytywane są ze standardowego wejścia zgodnie z poniższą specyfikacją. Pierwszą podawaną wartością będzie dodatnia liczba całkowita mniejsza od  $2^{15}$  oznaczająca ilość zestawów danych, po której na wejściu pojawią się zestawy danych w ilości równej wczytanej liczbie. Każdy zestaw danych zawiera w pierwszej linii dodatnią liczbę całkowitą  $n$ , przy czym  $3 \leq n \leq 100$  oznaczającą ilość odcinków wczytywanego zestawu. Zasadnicze dane zestawu w ilości równej poprzednio wczytanej wartości, będące liczbami całkowitymi dodatnimi.

Ostatnia linia każdego zestawu zakończona jest znakiem '\n'.

## Wyjście

Dla każdego zestawu danych wypisz:

- (a) w pierwszym wierszu numer zestawu i liczbę odcinków,
- (a) w kolejnych wierszach uporządkowaną niemalejąco tablicę odcinków po 25 elementów w jednym wierszu,
- (b) jeśli można z podanych odcinków zbudować co najmniej jeden trójkąt wypisz w nowym wierszu listę uporządkowanych leksykograficznie co najwyżej 10 pierwszych trójkątów, zawierających indeksy odcinków uporządkowanej tablicy, z których można zbudować trójkąt a w ostatnim wierszu wypisz liczbę wszystkich możliwych takich trójkątów indeksów tablicy np. Total number of triangles is: 15.

W przeciwnym przypadku wypisz zdanie: Total number of triangles is: 0.

## Wymagania implementacyjne

Jak programie P\_01.

Dane przykładowe

wejście: 5 4 2 3 2 3 6 1 2 1 2 3 3 6 1 4 2 2 3 3 4 1 1 1 1 4 1 3 1 4	wyjście: 1: n= 4 2 2 3 3 (0,1,2) (0,1,3) (0,2,3) (1,2,3) Total number of triangles is: 4 2: n= 6 1 1 2 2 3 3 (0,2,3) (0,4,5) (1,2,3) (1,4,5) (2,3,4) (2,3,5) (2,4,5) (3,4,5) Total number of triangles is: 8 3: n= 6 1 2 2 3 3 4 (0,1,2) (0,3,4) (1,2,3) (1,2,4) (1,3,4) (1,3,5) (1,4,5) (2,3,4) (2,3,5) (2,4,5) Total number of triangles is: 11 4: n= 4 1 1 1 1 (0,1,2) (0,1,3) (0,2,3) (1,2,3) Total number of triangles is: 4 5: n= 4 1 1 3 4 Total number of triangles is: 0
---	--